Technische Universität München
Prof. J. Esparza / Dr. M. Luttenberger, S. Sickert

Sommer 2016
27. Juni 2016

HA-Lösung

TA-Lösung

Einführung in die theoretische Informatik – Aufgabenblatt 9

Beachten Sie: Soweit nicht explizit angegeben, sind Ergebnisse stets zu begründen!

Hausaufgaben: Abgabe bis zum 22.06.2016 (Mittwoch) um 12:00

Aufgabe 9.1 Quiz — 1P+1P+1P+1P

Beantworten Sie die folgenden Fragen und geben Sie eine kurze Begründung an:

(a) Gibt es eine Turingmaschine, die den Kopf nie weiter als vier Schritte von der Startposition weg bewegt und eine unendliche Sprache akzeptiert?

(b) Welche Sprachen lassen sich mit Turingmaschinen, die ihren Kopf immer nur nach rechts bewegen, erkennen?

(c) Welche Sprachen lassen sich mit Turingmaschinen, die ihren Kopf immer nur nach links bewegen, erkennen?

(d) Kann jede Turingmaschine in eine Turingmaschine, die die gleiche Sprache erkennt und nur einen Zustand hat, übersetzt werden?

Lösung:

(a) Ja. Sei $q_0 \in F$, dann erkennt die TM Σ^*.

(b) Da kein geschriebenes Zeichen wieder gelesen werden kann, kann die TM von einem NFA simuliert werden: Reguläre Sprachen.

(c) Nur das erste Zeichen kann gelesen werden. Formal $\mathcal{L} = \{A\Sigma^* \cup B \mid A \subseteq \Sigma, B \subseteq \{\varepsilon\}\}$.

(d) Nein. Entweder $q_0 \in F$, dann erkennt die TM Σ^*, oder $q_0 \notin F$, dann erkennt die TM $\emptyset$.

Aufgabe 9.2 Produktkonstruktion für Kellerautomaten — 3P

Geben Sie eine direkte Konstruktion an, um aus einem PDA A und einem NFA N einen PDA A' mit $L_\varepsilon(A') = L_\varepsilon(A) \cap L(N)$ zu konstruieren. Verwenden Sie hierfür *nicht* den Umweg über kontextfreie Grammatiken.

Lösung: OBdA habe A nur einen Kontrollzustand q_A. Dann kann man in der Kontrolle von A einfach den NFA mitlesen lassen:

$$\delta_{A'}(p, a, X) = \{(q, \gamma) \mid q \in \delta_N(p, a) \wedge (q_A, \gamma) \in \delta_{A'}(q_A, a, X)\}$$

Einzige Schwierigkeit: A' soll mit leerem Keller akzeptieren, allerdings nur, wenn sich N in einem Endzustand befindet. Sei daher $\perp \notin \Gamma$ und $q_0 \notin Q_N$ und $q_A Z_0$ die eigentliche Startkonfiguration von A und p_0 der Startzustand von N. Dann ist $q_0\perp$ die Startkonfiguration von A' mit:

$$\delta_{A'}(q_0, \varepsilon, \perp) = \{(p_0, Z_0)\} \qquad \delta_{A'}(q_f, \varepsilon, \perp) = \{(q_f, \varepsilon)\}\,(q_f \in F_N)$$

A kennt $\perp$ nicht und kann daher $\perp$ nicht umschreiben. Damit kann $\perp$ nur von A' entfernt werden und das nur, wenn sich N in einem Endzustand befindet und der Stack von A leer ist. Ein Wort kann von A' somit höchstens dann akzeptiert werden, wenn sowohl A als auch N es akzeptieren. Formal zeigt man nun analog zur Produktkonstruktion für NFA, dass sich akzeptierende Abläufe von A und N zu einem akzeptierenden Ablauf von A' kombinieren lassen, und jeder akzeptierende Ablauf von A' akzeptierende Abläufe von A und N induziert. Zumindest Beweisskizze wird erwartet.

(Allgemeiner: Produktkonstruktion in Kontrolle.)

Aufgabe 9.3 **PDAs mit beschränktem Stack** **3P**

Zeigen Sie: Ist A ein PDA und $c \in \mathbb{N}_0$ eine Konstante, so dass es zu jedem $w \in L_\varepsilon(A)$ einen akzeptierenden Ablauf gibt, bei dem maximal c Symbol auf dem Stack liegen, dann ist $L_\varepsilon(A)$ regulär.

Lösung: Es reicht einen NFA N anzugeben, der nur die Abläufe des PDA simulieren kann, bei welchen zu jedem Zeitpunkt maximal c Symbole auf dem Stack liegen, was aber trivial ist, da man hierfür nur endlich viel Speicher braucht. $Q_N = Q_A \times \Gamma^{\leq c}$, kurz $q\gamma$ für $(q, \gamma) \in Q_A \times \Gamma^{\leq c}$, $F_N = Q_A$ und q_0 die Startkonfiguration von A. Damit:

$$\delta_N(pX\gamma', a) = \{q\gamma\gamma' \mid (q, \gamma) \in \delta_A(p, a, X)\} \cap Q_N$$

(Schnitt mit Q_N nötig, um Stacks mit mehr als c Symbolen rauszufiltern.)

Beweisskizze:

Akzeptiert A, dann gibt es nach Voraussetzung eine akzeptierende Berechnung, so dass zu jedem Zeitpunkt maximal c Symbole auf dem Stack sind, welche sich direkt in einen akzeptierenden Ablauf des konstruierten NFAs übersetzt, indem man einfach Zustand und Stackinhalt des PDA konkateniert, um den äquivalenten Zustand von N zu erhalten. Entsprechend übersetzt man auch jede akzeptierende Berechnung von N in eine akzeptierende Berechnung von A.

Aufgabe 9.4 **La-Ola-Wellen (Euro 2016)** **3P**

Wir konstruieren eine TM, die La-Ola-Wellen simuliert. Geben Sie hierzu eine deterministische TM an, welche als Eingabe ein Wort $w \in \{u, m, o\}^*$ mit $|w| \geq 3$ erwartet, wobei $w = a_0 \dots a_{l-1}$ den aktuellen Zustand einer La-Ola-Welle beschreibt, wobei sich die Welle bei a_{l-1} wieder bei a_0 fortsetzen sollen (mod$l$). Die Buchstaben von w beschreiben die aktuelle Armhaltung (**u**nten, **m**ittig, **o**ben) des Zuschauers auf Platz i.

Die DTM soll zuerst prüfen, dass w eine zulässige La-Ola-Welle ist, d.h.

$$\begin{array}{rcl} & & (a_i = o \to a_{i-1 \bmod l} a_i a_{i+1 \bmod l} \in L((o|m)o(o|m))) \\ \forall i \in \mathbb{Z}_l\colon & \wedge & (a_i = u \to a_{i-1 \bmod l} a_i a_{i+1 \bmod l} \in L((u|m)u(u|m))) \\ & \wedge & (a_i = m \to a_{i-1 \bmod l} a_i a_{i+1 \bmod l} \in L(umo|omu)) \end{array}$$

und anschließend die Welle um eine Position nach links verschieben, d.h. falls w zulässig ist, soll die DTM mit der Ausgabe $\triangleright a_1 \dots a_{l-1} a_0 \triangleleft$ terminieren. Falls die Eingabe jedoch nicht zulässig ist, soll die DTM mit leerem Band terminieren.

Lösung: Zur Vereinfachung prüft man zunächst nur, dass die Eingabe die korrekte Struktur hat, hierzu speichert man in der Kontrolle z.B. a_0, a_{i-2}, a_{i-1}, um entscheiden zu können, ob a_i korrekt ist und ob a_{i-1} korrekt ist. Sei $L_A = L((o|m)o(o|m)) \cup L((u|m)u(u|m)) \cup L(umo|omu)$.

Speicher initialisieren, wobei (x, x', y, z) in x das Zeichen a_0, in x' das Zeichen a_1, in y das Zeichen a_{i-2} und in z das Zeichen a_{i-1} speichert:

$$(\varepsilon, \varepsilon, \varepsilon, \varepsilon) \xrightarrow[x \in \{u,o,m\}]{x/x,R} (x, \varepsilon, x, \varepsilon) \qquad (x, \varepsilon, x, \varepsilon) \xrightarrow[x' \in \{u,o,m\}]{x'/x',R} (x, x', x, x') \qquad (x, \varepsilon, x, \varepsilon) \xrightarrow{\square/\square,L} \mathsf{del}$$

Auf Zulässigkeit testen an Positionen $i = 1, \dots, l-2$:

$$(x, x', y, z) \xrightarrow[(y,z,a) \in L_A]{a/a,R} (x, x', z, a) \qquad (x, x', y, z) \xrightarrow[(y,z,a) \notin L_A]{a/a,R} \mathsf{mvr}$$

Auf Zulässigkeit testen an Positionen $i = 0$ und $i = l-1$:

$$(x, x', y, z) \xrightarrow[(y,z,x) \in L_A \wedge (z,x,x') \in L_A]{\square/\square,L} (x) \qquad (x, x', y, z) \xrightarrow[(z,a,x) \notin L_A \vee (z,x,z') \notin L_A]{\square/\square,L} \mathsf{del}$$

Eingabe um eine Positionen nach Links shiften:

$$(x) \xrightarrow[x,y \in \{u,m,o\}]{y/x,L} (y) \qquad (x) \xrightarrow[x,y \in \{u,m,o\}]{\square/\square,R} \mathsf{fin}$$

Rechtes Ende suchen, um Eingabe zu löschen:

$$\mathsf{mvr} \xrightarrow[x \in \{u,o,m\}]{x/x,R} \mathsf{mvr} \qquad \mathsf{mvr} \xrightarrow{\square/\square,L} \mathsf{del}$$

Eingabe löschen:

$$\mathsf{del} \xrightarrow[x \in \{u,m,o\}]{x/\square,L} \mathsf{del} \qquad \mathsf{del} \xrightarrow{\square/\square,N} \mathsf{del}$$

Tutoraufgaben: Besprechung in KW25

Aufgabe 9.1 **Turing Maschinen**

Geben Sie für jede der angegebenen Sprachen eine TM an, welche diese entscheidet:

$$L_1 = \{a^n b^n c^n \mid n \in \mathbb{N}_0\} \qquad L_2 = \{a^n b^{n^2} \mid n \in \mathbb{N}_0\}$$

Lösung: Wir schreiben $\square$ für eine leere Bandzelle.

- Sei TM $M_1 = (\{q_0, p_a, p_b, p_c, p_L, q_a, q_b, q_c, q_d, q_L, q_F\}, \{a, b, c\}, \Sigma \cup \{c, \square\}, \delta, q_0, \square, \{q_F\})$.

  Terminiere sofort bei leerer Eingabe:

  $$q_0 \xrightarrow{\square/\square,N} q_F \qquad q_0 \xrightarrow{a/a,N} p_a$$

  Überprüfe erst, dass die Eingabe die geforderte Struktur $a^*b^*c^*$ hat. (Prinzipiell unnötig, lässt sich auch später einbauen bzw. wird dort eigentlich automatisch geprüft.)

  $$p_a \xrightarrow{a/a,R} p_a \qquad p_a \xrightarrow{b/b,R} p_b \qquad p_b \xrightarrow{b/b,R} p_b \qquad p_b \xrightarrow{c/c,R} p_c \qquad p_c \xrightarrow{c/c,R} p_c \qquad p_c \xrightarrow{\square/\square,L} p_L$$

  Laufe dann zum ersten a von rechts.

  $$p_L \xrightarrow{b/b|c/c,L} p_L \qquad p_L \xrightarrow{a/a,N} q_a$$

  Wiederhole: Ersetze rechtestes a durch d; laufe über alle d bis zum ersten b und ersetze dieses durch d;

  $$q_a \xrightarrow{a/d,R} q_b \qquad q_b \xrightarrow{d/d,R} q_b \qquad q_b \xrightarrow{b/d,R} q_c$$

  laufe bis zum ersten c über alle b und d und ersetze dieses durch d;

  $$q_c \xrightarrow{b/b|d/d,R} q_c \qquad q_c \xrightarrow{c/d,L} q_L$$

  laufe zum ersten a nach ganz links;

  $$q_L \xrightarrow{b/b,c/c,d/d,L} q_L \qquad q_L \xrightarrow{a/a,N} q_a \qquad q_L \xrightarrow{\square/\square,R} q_d$$

  Wenn keine a mehr vorhanden, überprüfe, dass die Eingabe nur noch aus d besteht:

  $$q_d \xrightarrow{d/d,R} q_d \qquad q_d \xrightarrow{\square/\square,N} q_F$$

- Idee: Benutze den a-Bereich auch als Zähler.

  (Alternativ: Schreibe b^{n^2} zu $b^{(n-1)^2}$ um, in dem man ein a und $2n-1$ b löscht.)

  (a) Terminiere sofort bei leerer Eingabe, sonst gehe zu (c):

  $$q_0 \xrightarrow{\square/\square,N} f \qquad q_0 \xrightarrow{a/a,N} q$$

  (b) Laufe einmal bis ganz nach links und lösche alle Markierungen auf dem Weg dorthin

  $$p \xrightarrow{a'/a|d'/d,L} p \qquad p \xrightarrow{\square/\square,R} q$$

  (c) Laufe zum ersten a von links und lösche es:

  $$q \xrightarrow{d/d,R} q \qquad q \xrightarrow{a/d,L} r$$

  (d) Laufe wieder zum linkesten Symbol:

  $$r \xrightarrow{d/d,L} r \qquad r \xrightarrow{\square/\square,R} s$$

  (e) Suche erstes a bzw. d von links und markiere es:

  $$s \xrightarrow{d'/d'|a'/a',R} s \qquad s \xrightarrow{d/d'|a/a',L} t$$

  (f) Suche rechtestes b:

  $$t \xrightarrow{a/a|b/b,R} t \qquad t \xrightarrow{\square/\square,L} u$$

(g) Lösche es:

$$u \xrightarrow{b/\square,R} v$$

(h) Suche rechtestes a' bzw. d':

$$v \xrightarrow{b/b|a/a|d/d,L} v \qquad v \xrightarrow{a'/a'|d'/d',R} w$$

Falls noch unmarkietes a oder d übrig, markiere wechsle in Zustand t (goto (e)), falls noch ein b übrig ist, wechsle in Zustand p (goto (a))

$$w \xrightarrow{a/a'|d/d',L} t \qquad w \xrightarrow{b/b,R} p \qquad w \xrightarrow{\square/\square} x$$

(i) ansonsten akzeptiere, falls Eingabe nur noch aus markierten d' besteht:

$$x \xrightarrow{d'/d',N} f$$

Aufgabe 9.2 NFA, PDA → TM

Beschreiben Sie allgemein, wie man die Beschreibung $N = (Q_N, \Sigma_N, \delta_N, q_N, F_N)$ eines NFA bzw. eines PDA $P = (Q_P, \Sigma_P, \Gamma_P, \delta_P, q_P,$ in die Beschreibung M einer TM überführt, so dass T gerade $L(N)$ bzw. $L_\varepsilon(P)$ entscheidet. Verwenden Sie bei Bedarf eine TM mit mehreren Bändern.

Lösung:

(a) NFA zu TM: TM liest einfach Eingabe von links nach rechts und vollzieht nicht deterministisch die Zustandswechsel, die der NFA für das gelesene Zeichen machen würde. Bei Erreichen des rechten Rands der Eingabe, stoppt die TM und akzeptiert, falls der NFA sich in einem Endzustand befindet: (oBdA. $f \notin Q_N$)

$$Q_M = Q_N \qquad \Sigma_M = \Sigma_N \qquad \Gamma_M = \Sigma_M \cup \{\square\} \qquad F_M = \{f\} \qquad q_M = q_N$$

und

$$\delta_M(p,a) = \{(q,a,R) \mid (p,a,q) \in \delta_N\} \qquad \delta_M(q,\square) = \{(f,\square,N)\} \text{ falls } q \in F_N$$

(b) PDA zu TM: Wir nehmen an, dass der PDA mit leerem Keller akzeptiert und ein explizites Bottom-Symbol $\perp$ verwendet. Die TM simuliert den Stack des PDA auf einem zweiten Band, wobei das zweite Band oBdA. zu Beginn mit $\perp$ initialisiert ist. Das rechtestes Symbol des zweiten Bandes entspricht dann dem obersten Symbol auf dem Stack des simulierten PDA. Damit:

$$\delta_M(p,a,X) = \{(q,a,Y,R,N) \mid pX \xrightarrow{a} qY\} \cup \{(q,a,\square,R,L) \mid pX \xrightarrow{a} q\} \cup \{((q,Y),a,Z,R,R) \mid pX \xrightarrow{a} qYZ\}$$

$$\delta_M(p,a,X) = \{(q,a,Y,N,N) \mid pX \xrightarrow{\varepsilon} qY, a \in \Gamma\} \cup \{(q,a,\square,N,L) \mid pX \xrightarrow{\varepsilon} q, a \in \Gamma\} \cup \{((q,Y),a,Z,N,R) \mid pX \xrightarrow{\varepsilon} qYZ, a \in \Gamma\}$$

zzgl.

$$\delta((q,Y),a,\square) = \{(q,a,Y,N,N) \mid (q,Y) \in Q_M, a \in \Gamma\}$$

und

$$\delta(q,\square,\square) = \{(f,\square,\square)\}$$

wobei $Q_M = Q_P \cup Q_P \times \Gamma_P$.

Aufgabe 9.3 k-PDA

Ein k-PDA ist ein PDA, der k Stacks zur Verfügung hat. In jedem Schritt, kann der PDA in Abhängigkeit vom aktuellen Zustand, dem gelesenen Eingabezeichen und den Symbolen, die zu oberst auf jedem der k Stacks liegen, in einen neuen Zustand wechseln und jeden der k Stacks wie im Fall eines gewöhnlichen PDAs modifizieren. Offensichtlich ist jeder k-PDA auch ein $k+1$-PDA.

Zeigen bzw. argumentieren Sie:

(a) 0-PDA und NFA akzeptieren dieselben Sprachen.

(b) Es gibt eine Sprache, die von einem 2-PDA, aber von keinem 1-PDA akzeptiert wird.

(c) Für jeden k-PDA ($k \geq 3$) gibt es einen 2-PDA, der dieselbe Sprache akzeptiert, d.h. mehrere Stacks können immer durch genau zwei Stacks simuliert werden.

Lösung:

(a) Ein 0-PDA kann nur Zustandsänderungen vollziehen, aber nicht auf dem Stack schreiben. Die Definitionen von 0-PDA und NFA fallen damit zusammen.

(b) Ein 2-PDA kann die Sprache $a^n b^n c^n$ akzeptieren, indem er zuerst alle as auf den ersten, dann alle bs auf den zweiten Stack schiebt, dann für jedes c jeweils ein Zeichen von beiden Stacks popt und schließlich nur dann akzeptiert, falls nach Lesen der Eingabe beide Stacks leer sind.

(c) Jeder 2-PDA kann eine TM mit einem Band simulieren, indem der erste Stack alle Symbole links vom Lesekopf, der zweite Stack alle Symbole rechts vom Lesekopf speichert.

Eine k-Band-TM kann jeden k-PDA simulieren.

Jede k-Band-TM kann von einer 1-Band-TM simuliert werden.

Damit kann jeder 2-PDA jeden k-PDA simulieren und ist so mächtig wie jede TM.